

# Centro de Investigación y Estudios Avanzados del I.P.N.



## Reconocimiento de 8 herramientas

Materia: Analisis de imagenes digitales

Catedratico: Dr. Wilfrido Gomez-

Alumno: Luis Alberto Ballado Aradias

27 de agosto de 2023

# Índice

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Resultados</b>	<b>7</b>
2.1	Umbralado con el método de Otsu . . . . .	7
2.2	Superpixels . . . . .	7
2.3	K-means . . . . .	7
2.4	Descripción con rasgos de forma . . . . .	7
2.4.1	Área y perímetro . . . . .	8
2.4.2	Rasgos basados en esqueleto . . . . .	8
2.4.3	Momentos invariantes . . . . .	8
<b>3</b>	<b>Clasificación de objetos</b>	<b>9</b>
<b>4</b>	<b>Apendice</b>	<b>10</b>
4.1	Script de rasgos . . . . .	10
4.2	Script de data augmentation . . . . .	10
<b>5</b>	<b>Experimentando en python</b>	<b>12</b>
<b>6</b>	<b>Conclusiones</b>	<b>13</b>

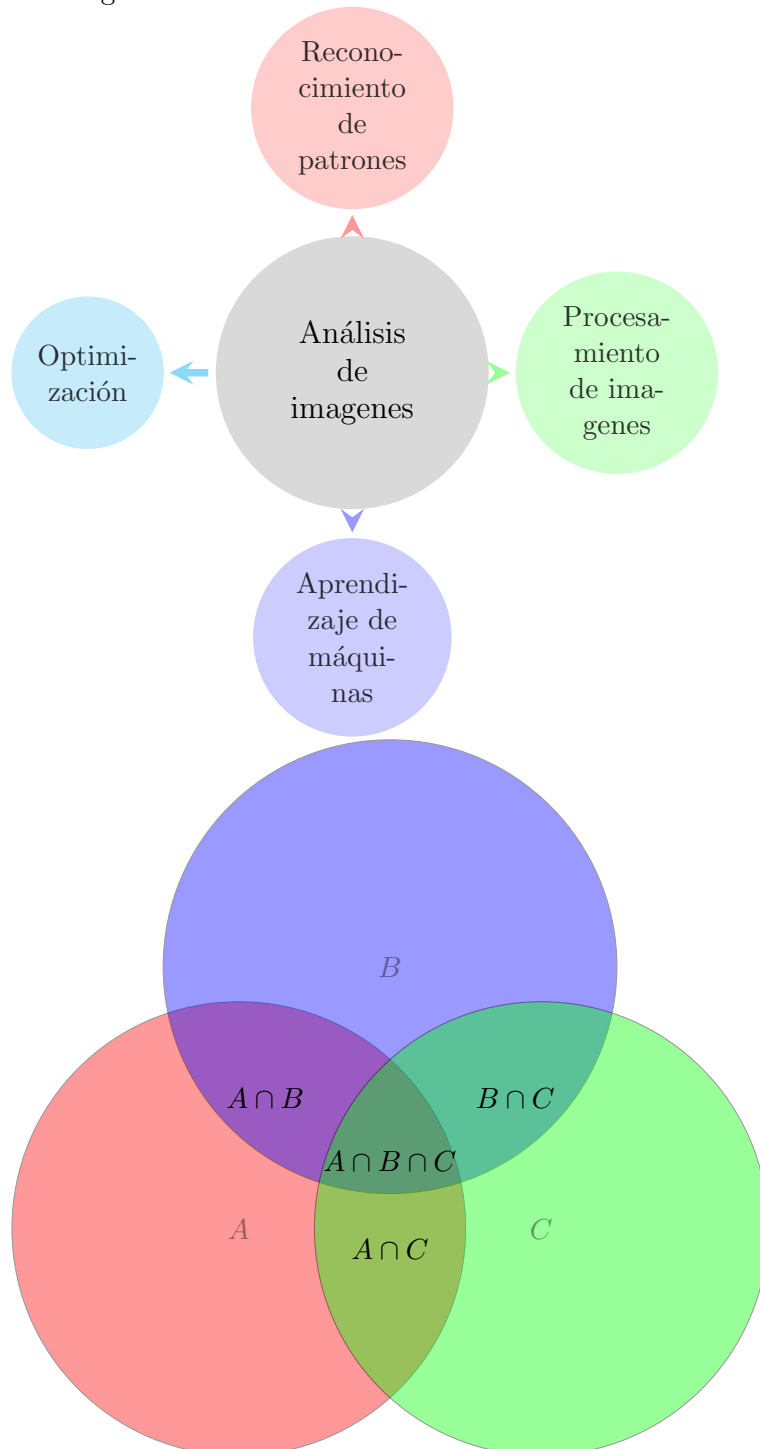
List of Code

1	Python example . . . . .	10
2	Python example . . . . .	10

## 1. Introducción

El Análisis de imágenes digitales es un área fascinante que engloba muchas áreas de las ciencias computacionales como Optimización y Aprendizaje de máquinas.

La imagen muestran las áreas dentro de los Análisis de imágenes



Las dificultades que se tienen dentro del procesamiento de imágenes digitales son la iluminación del ambiente, siendo los sensores perceptibles a la iluminancia al obtener la imagen.

Una imagen  $f(x, y)$  se puede modelar como el producto de dos componentes  $f(x, y) = i(x, y) \cdot r(x, y)$ , donde  $0 < i(x, y) < \infty$  es la componente de iluminación y  $0 < r(x, y) < 1$  es la componente de reflectancia.

Las operaciones aritméticas entre píxeles, la adición de la misma imagen por varias iteraciones reduce el ruido, sustracciones de dos imágenes de la misma toma detecta los cambios en los píxeles de la imagen, la multiplicación de imágenes para enmascarar zonas de interés.

En la información que puede obtener de sus vecinos para detectar bordes a partir de distancias entre píxeles.



Figura 1: Example of a parametric plot  $(\sin(x), \cos(x), x)$

Euclidiana  $D_e(p, q) = [((x - u)^2 + (y - v)^2)^{\frac{1}{2}}]$

cuatro vecindad usando los vecinos horizontales y verticales.

Suponga una imagen con  $k$  regiones disjuntas  $R_k, k = 1, 2, \dots, k$  sea  $R_u$  la unión de las  $k$  regiones y  $(R_u)^c$  su complemento.

Entonces los puntos en  $R_u$  representan el primer plano y los puntos de  $(R_u)^c$  representan el fondo.

Contorno de una region

El contorno de una región conexa es el conjunto de píxeles que tienen al menos un pixel vecino que corresponde al fondo en 4 ó 8 adyacencia.

Buscar que tipo de etiquetado usa BWLABEL .- QUE ALGORITMO?

Etiquetado de regiones conexas

Data augmentation

Se usaron transformaciones geometricas para modificar la relación espacial entre píxeles en una imagen.

Usando propiedades que no deformen al objeto de interes.

- Rotaciones
- Traslaciones

Logrando crear imagenes artificiales.

El mejoramiento como manipulación de una imagen tal que el resultado sea más útil que la imagen original para una aplicación particular.

Una transformación de intensidad modifica el contraste.

Ecualización del histograma.

No se tuvo que ecualizar ni mejorar por CLAHE, pero es bueno saber la existencia de estas tecnicas para crear sistemas y filtros sofisticados.

Modo de degradación de una imagen.

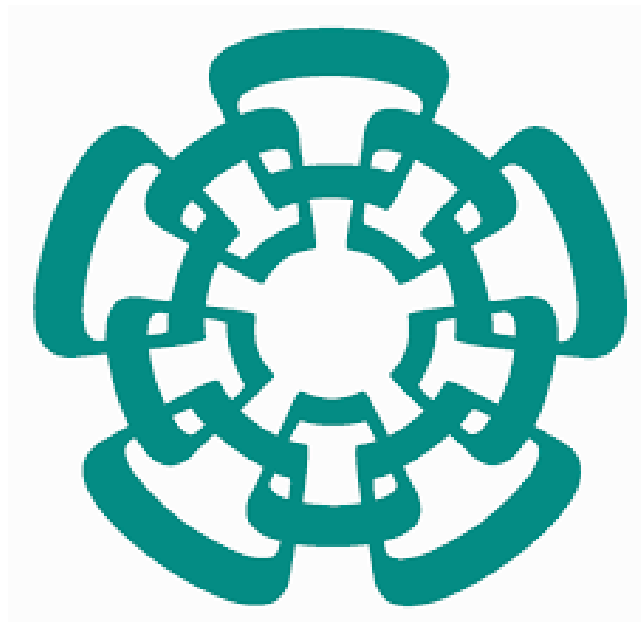


Figura 2: Modelos de ruido

No se hizo uso de ningun filtro, buscando evitar el efecto de deslavado y poder detectar los bordes lo más puros posibles.

Pero el filtro de difusión anisotrópico puede considerarse en la implementación.

Se probó la detección de bordes a partir del gradiente, pero la demasiada información nos afectaba en un ciclo para detectar bordes como canny edges.

## EJEMPLO DE LAPLACIANO DEL GAUSIANO

Los beneficios de la aplicación del teorema de convolución nos ayuda en la reducción de la complejidad computacional, al solo considerar una multiplicación en frecuencia.

### Operadores morfologicos

Los elementos estructurantes, donde podemos encontrar la línea puede tener inclinaciones a ciertos grados y ser de 4 vecindad ó 8 vecindad.

La erosión y dilatación que son operadores muy buenos y fueron aplicados en el proyecto.

- Apertura: una erosión seguida de una dilatación  $fob = (f - b) + b$
- Cerradura: una dilatación, seguida de una erosión  $fob = (f + b) - b$

El borde de un conjunto A que contiene los píxeles del primer plano se obtiene como:

$B(A) = A - (A - b)$  donde A es la figura binaria y b el elemento estructurante.

Rellenado de orificios.

Un orificio es una región del fondo rodeado por un borde conexo de pixeles del primer plano.

Se efectua una reconstrucción por dilatación.

## 2. Resultados



Figura 3: Esqueletización

Dentro de las imagenes, la similitud de esqueletos entre la cinta, la llave perica, tijera, pinzas de punta y electrica que tienen formas similares.

La mayoría de desarmadores no tiene ramas y tiene dos puntos finales, lo que lo confunde con una cinta al solo considerar las ramas y los puntos terminales se tiene una exactitud del 20 %, mientras al agregar el num de pixeles crece al 78 %, agregando más información como el área y perimetro la probabilidad de acierto crece aún más, presentandose valores arriba del 90 %.

### 2.1. Umbralado con el método de Otsu

Al afectarnos las partes cromadas, se busco crear una tecnica basica pero asu vez robusta a cambios de colores haciendo uso de bordes de canny.

### 2.2. Superpixels

### 2.3. K-means

### 2.4. Descripción con rasgos de forma

El reconocimiento automático de objetos requiere calcular rasgos que describan propiedades fisicas de los objetos.

Un rasgo (atributo o característica), es un valor numérico que cuantifica alguna propiedad de forma, textura, color, geometria, etc.

- Un rasgo debe ser discriminante, invariante, incorrelado y rápido de computar.
- Los rasgos de forma generalmente se calculan a partir de la segmentación del objeto y se divide entre todos los pixeles de las regiones de interes.



- Basados en región distribución de todos los píxeles.
- Basados en contorno, variaciones a lo largo del contorno.

#### **2.4.1. Área y perímetro**

El área es el número total de píxeles que cubre la región del objeto.

El perímetro es la longitud del contorno del objeto. El resultado depende del tipo de conectividad que pueda tener.

Elementos básicos.

- Cerco convexo
  - Solidez
  - Convexidad
- Esqueleto
- Geométricos
  - Redondez
  - Circularidad
  - Compacidad
  - Factor de forma

#### **2.4.2. Rasgos basados en esqueleto**

- Puntos finales
- Ramas
- Número de píxeles

#### **2.4.3. Momentos invariantes**

Los momentos son proyecciones de una función sobre una base polinomial usados para medir la distribución de masa de un cuerpo.

### 3. Clasificación de objetos

Capas del reconocimiento de objetos en imágenes.

1. Sensado.- Capturar la imagen.
2. Preprocesamiento.- Mejorar la calidad de la imagen y segmentar los objetos de interés.
3. Extracción de características.- Describir los objetos con rasgos cuantitativos discriminantes e invariantes.
4. Clasificación.- Asignar una etiqueta de clase a cada objeto.

Obtener un vector de patrones.

Para el proyecto se consideran ocho clases siendo representadas por los objetos siguientes:

1. Cinta de medir
2. Desarmador
3. Martillo
4. Pinza eléctrica
5. Pinza de presión
6. Pinza de punta
7. Perica
8. Tijeras

En este proyecto se usa el método de datos de entrenamiento que son vectores de patrones asociados a una etiqueta de clase  $(x,y)$  con  $y \in \Omega = w_1, \dots, w_c$

MOSTRAR EN DIAGRAMA DE LOS PROCESOS PARA LA CLASIFICACION

La selección del algoritmo de clasificación debe tener las siguientes cualidades.

- Generación de fronteras de decisión no lineales.
- Clasificación en más de dos clases (multiclases).
- Entrenamiento en un tiempo de cómputo razonable.

Se elige el clasificador k-nn de vecinos más cercanos por su simpleza.

## 4. Apendice

### 4.1. Script de rasgos

We did some experiments ...

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
20                VT[(j)*n + c[k]] = 1;
21
22            if M is None:
23                M = np.copy(VT)
24            else:
25                M = np.concatenate((M, VT), 1)
26
27            VT = np.zeros((n*m,1), int)
28
29    return M
```

Listing 1: Python example

### 4.2. Script de data aumentation

We did some experiments ...

```
1 import numpy as np
2
3 def incmatrix(genl1,genl2):
4     m = len(genl1)
5     n = len(genl2)
6     M = None #to become the incidence matrix
7     VT = np.zeros((n*m,1), int) #dummy variable
8
9     #compute the bitwise xor matrix
10    M1 = bitxormatrix(genl1)
11    M2 = np.triu(bitxormatrix(genl2),1)
12
13    for i in range(m-1):
14        for j in range(i+1, m):
15            [r,c] = np.where(M2 == M1[i,j])
16            for k in range(len(r)):
17                VT[(i)*n + r[k]] = 1;
18                VT[(i)*n + c[k]] = 1;
19                VT[(j)*n + r[k]] = 1;
```

```

20         VT[(j)*n + c[k]] = 1;
21
22         if M is None:
23             M = np.copy(VT)
24         else:
25             M = np.concatenate((M, VT), 1)
26
27         VT = np.zeros((n*m,1), int)
28
29     return M

```

Listing 2: Python example

## 5. Experimentando en python

From our experiments we [\[1\]](#) can conclude that ...

## 6. Conclusiones

Estudios tempranos de Inteligencia Artificial buscaban duplicar los pensamientos humanos, pero ahora los estudios recientes muestran la tendencia en replicar el resultado y las computadoras actúan como sistemas expertos.

Las computadoras son dispositivos simbólicos capaces de manipular cualquier tipo de símbolos siendo los números una clase importante, pero las computadoras son más generales que eso. Sabemos de la generalidad de la computación desde los tiempos de Alan Turing en los 1930's y se tienen intuiciones que Babbage tuvo también estudios que fueron afirmados por Ada Lovelace, en 1842 Ada Lovelace escribió sobre la ingeniería analítica de Babbage que buscaba unir el mundo mecánico con el mundo de las cosas abstractas, en la psicología moderna es llamado The physical symbol system hypothesis y son las bases en que la Inteligencia Artificial funciona.

La Inteligencia Artificial como ciencia emplea el uso de computadoras para procesar conocimientos simbólicos usando métodos de inferencias lógicas, en otras palabras, nos referimos a inferencia y no cálculos que se piensa en el sentido tradicional, hablamos de conocimiento y no números como en la forma tradicional.

Al poner conocimiento en programas computacionales que para los humanos nos es fácil o a veces un reto intelectual y el conocimiento que le pasamos es representativo en cierta forma particular.

Por otra parte, los sistemas expertos basados en conocimientos son aplicables en cualquier área en que sea especializado el conocimiento y sea rutinario la toma de decisiones, estrategias para resolver problemas, diagnósticos, .. etc.

Siendo de gran ayuda en un gran rango de aplicaciones, el **conocimiento médico**, en clasificación de un experto, puede estar reflejado con el funcionamiento de una red neuronal.

Para algún diagnóstico médico o segmentadores más fáciles, pero igual de complicados como la segmentación en imágenes para la creación de sistemas autónomos.

Programas que se puedan ejecutar en cualquier computadora o dispositivo que permita la ejecución de código multiplataforma como python o C++, que tienen capacidades altas de manipulación simbólica, la toma de decisiones es primordial en una inteligencia basada en conocimiento y esa manipulación simbólica es necesaria.

## Referencias

- [1] B. Zhou, H. Xu, and S. Shen. Racer: Rapid collaborative exploration with a decentralized multi-uav system. *IEEE Transactions on Robotics*, 39(3):1816–1835, 2023.